Visualização de dados com ggplot2

Prof. Walmes Zeviani

walmes@ufpr.br

Laboratório de Estatística e Geoinformação
Departamento de Estatística
Universidade Federal do Paraná

Recursos gráficos do R

Visão geral

O landscape de recursos para visualização de dados no R pode ser representado por uma divisão em 4 territórios.

1. O pacote `graphics` e derivados.
2. O pacote `lattice` e derivados.
3. O pacote `ggplot2` e derivados.
4. Pacotes para gráficos interativos.

O pacote `graphics`

- Contém os recursos mais primitivos: com funções de alto e baixo nível.
- Pouco suporte para mapeamento em variáveis visuais retinais.
- Vários pacotes que complementam suas funcionalidades: `plotrix` e `gplots`.
- Usado em métodos gráficos de saídas de análises: dendrogramas, biplots, etc.

O pacote `lattice`

- Desenvolvido por Deepayan Sarkar.
- Plotagem multi-painel e mapeamento em variáveis visuais retinais.
- Já vem com a instalação básica do R.
- Também é utilizado na implementação de métodos gráficos.

O pacote `ggplot2`

- Desenvolvido por Hadley Wickham.
- Baseado na *Grammar of Graphics*.
- Plotagem multi-painel e mapeamento em variáveis visuais retinais.
- A importância da `ggplot2` está no modelo mental mais claro.

Pacotes para recursos interativos

A visualização interativa é voltada para exibicação na WEB. Alguns dos pacotes para isso são estes:

- `plotly`.
- `highcharter`.
- `googleVis`.
- `rCharts`.
- `leaflet`.
- `iplots`.
- `rgl`.
- `animation`.

Mais informação

Para mais detalhes sobre os recursos gráficos, siga esse link: https://www.stat.ubc.ca/~jenny/STAT545A/block90_baseLatticeGgplot2.html.

Para uma descrição completada comparação entre `lattice` e `ggplot2`, siga esse link: https://learnr.wordpress.com/2009/08/26/ggplot2-version-of-figures-in-lattice-multivariate-data-visualization-with-r-final-p

Confira no R Graph Gallery a variedade de gráficos confeccionados com o R.

Galerias e tutoriais

The R Graph Gallery

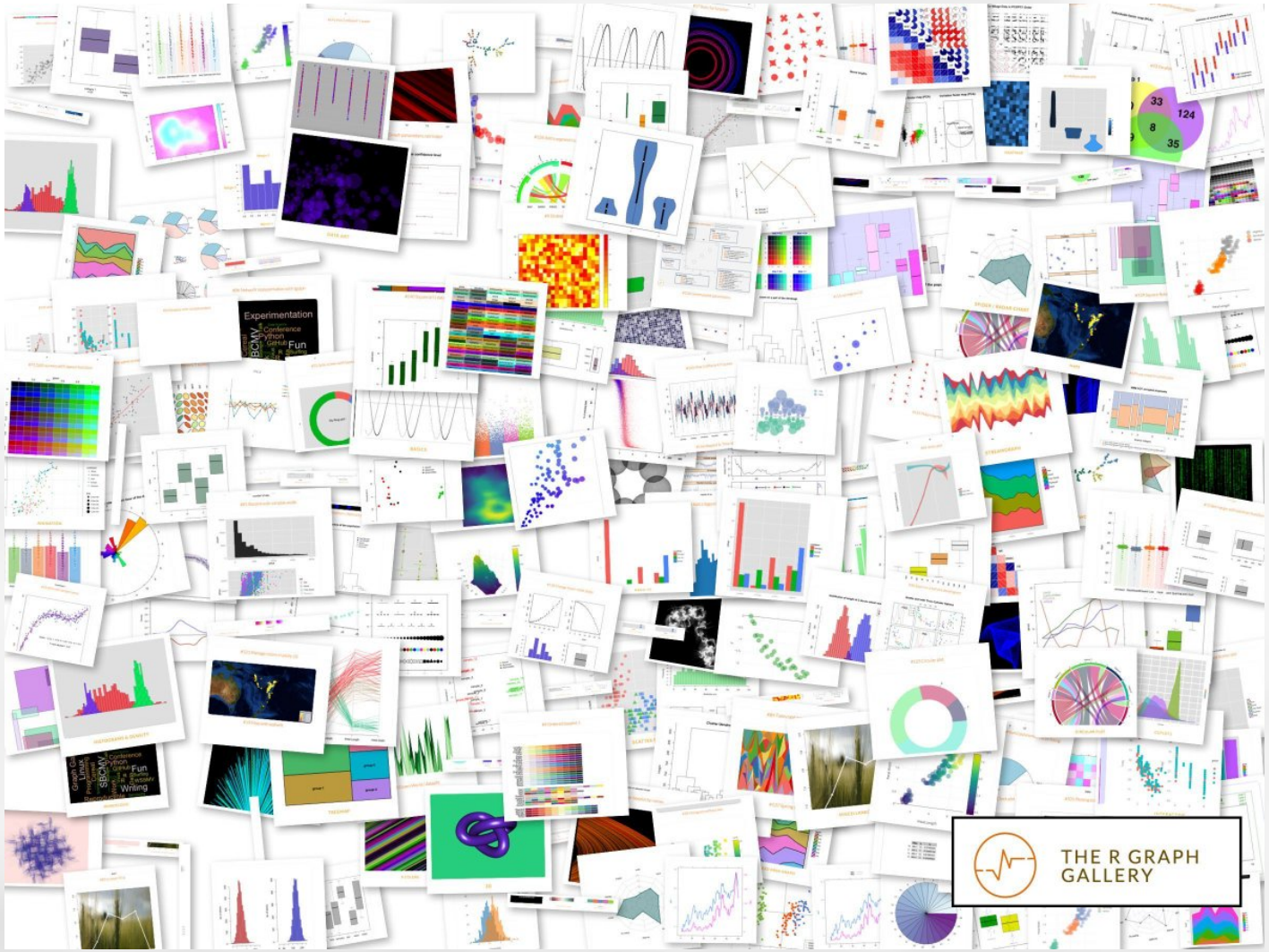

Figura 1. Gráficos da galeria de gráficos do R. Fonte: Mara Averick.

Galerias

- https://www.r-graph-gallery.com/portfolio/ggplot2-package/.
- http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html.
- https://r4stats.com/examples/graphics-ggplot2/.
- http://girke.bioinformatics.ucr.edu/GEN242/pages/mydoc/Rgraphics.html.

Extensões do ggplot2

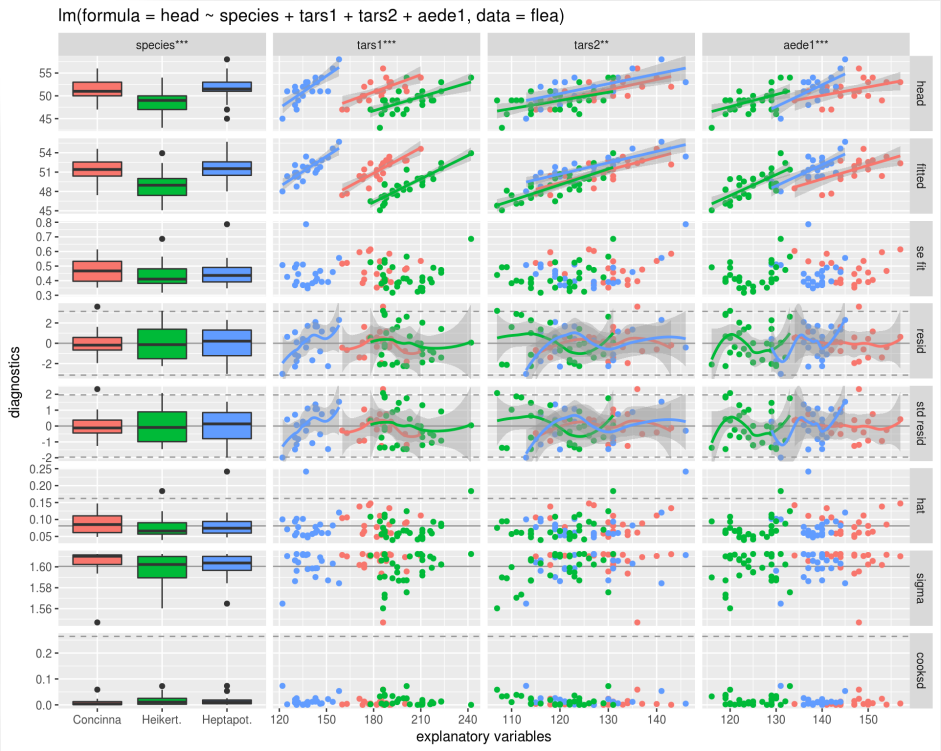

Figura 2. Gráficos da galeria de extensões do ggplot2. Fonte: GGally.

- http://www.ggplot2-exts.org/gallery/.
- https://mode.com/blog/r-ggplot-extension-packages.

Mapas

- https://rstudio-pubs-static.s3.amazonaws.com/176768_ec7fb4801e3a4772886d61e65885fbdd.html.
- https://www.curso-r.com/blog/2017-05-04-mapas-tematicos-3-minutos/.
- http://girke.bioinformatics.ucr.edu/GEN242/pages/mydoc/Rgraphics.html.

Tutoriais em português

- https://rpubs.com/mnunes/ggplot2.
- https://analisereal.com/2015/09/19/introducao-ao-ggplot2/.
- https://timogrossenbacher.ch/2016/12/beautiful-thematic-maps-with-ggplot2-only/.
- http://recologia.com.br/tag/graficos/.
- http://rstudio-pubs-static.s3.amazonaws.com/24563_3b7b0a6414824e3b91769a95309380f1.html.
- http://eduardogutierres.com/inteligencia-geografica-gerando-mapas-em-r/.
- https://pt.stackoverflow.com/questions/332053/r-mapa-de-cidades-brasileiras.

Um overview do `ggplot2`

A ficha técnica

ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics

A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Version:	3.1.0
Depends:	R (≥ 3.1)
Imports:	digest, grid, gtable (≥ 0.1.1), lazyeval, MASS, mgcv, plyr (≥ 1.7.1), reshape2, rlang (≥ 0.2.1), scales (≥ 0.5.0), stats, tibble, viridisLite, withr (≥ 2.0.0)
Suggests:	covr, dplyr, ggplot2movies, hexbin, Hmisc, lattice, mapproj, maps, maptools, multcomp, munsell, nlme, testthat (≥ 0.11.0), vdiffr, quantreg, knitr, rgeos, rpart, rmarkdown, sf (≥ 0.3-4), svglite (≥ 1.2.0.9001)
Enhances:	sp
Published:	2018-10-25
Author:	Hadley Wickham [aut, cre], Winston Chang [aut], Lionel Henry [aut], Thomas Lin Pedersen [aut], Kohske Takahashi [aut], Claus Wilke [aut], Kara Woo [aut], RStudio [cph]
Maintainer:	Hadley Wickham <hadley at rstudio.com>
BugReports:	https://github.com/tidyverse/ggplot2/issues
License:	GPL-2 | file LICENSE
URL:	http://ggplot2.tidyverse.org, https://github.com/tidyverse/ggplot2
NeedsCompilation:	no
Citation:	ggplot2 citation info
Materials:	README NEWS
In views:	Graphics, Phylogenetics, TeachingStatistics
CRAN checks:	ggplot2 results

Figura 3. Ficha técnica do `ggplot2`.

Gramática dos gráficos

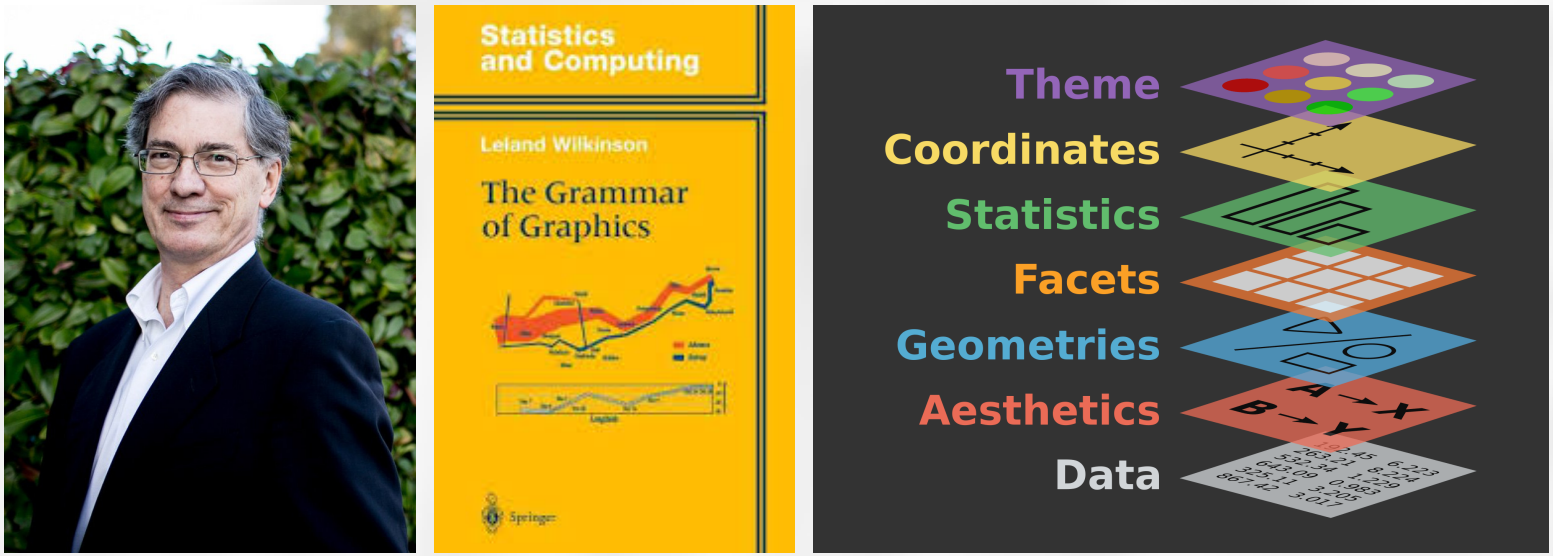

Figura 4. Leland Wilkinson (esq.) autor de “The grammar of graphics” (meio) e as camadas da gramática de gráficos que são usadas no `ggplot2`.

Gramática dos gráficos

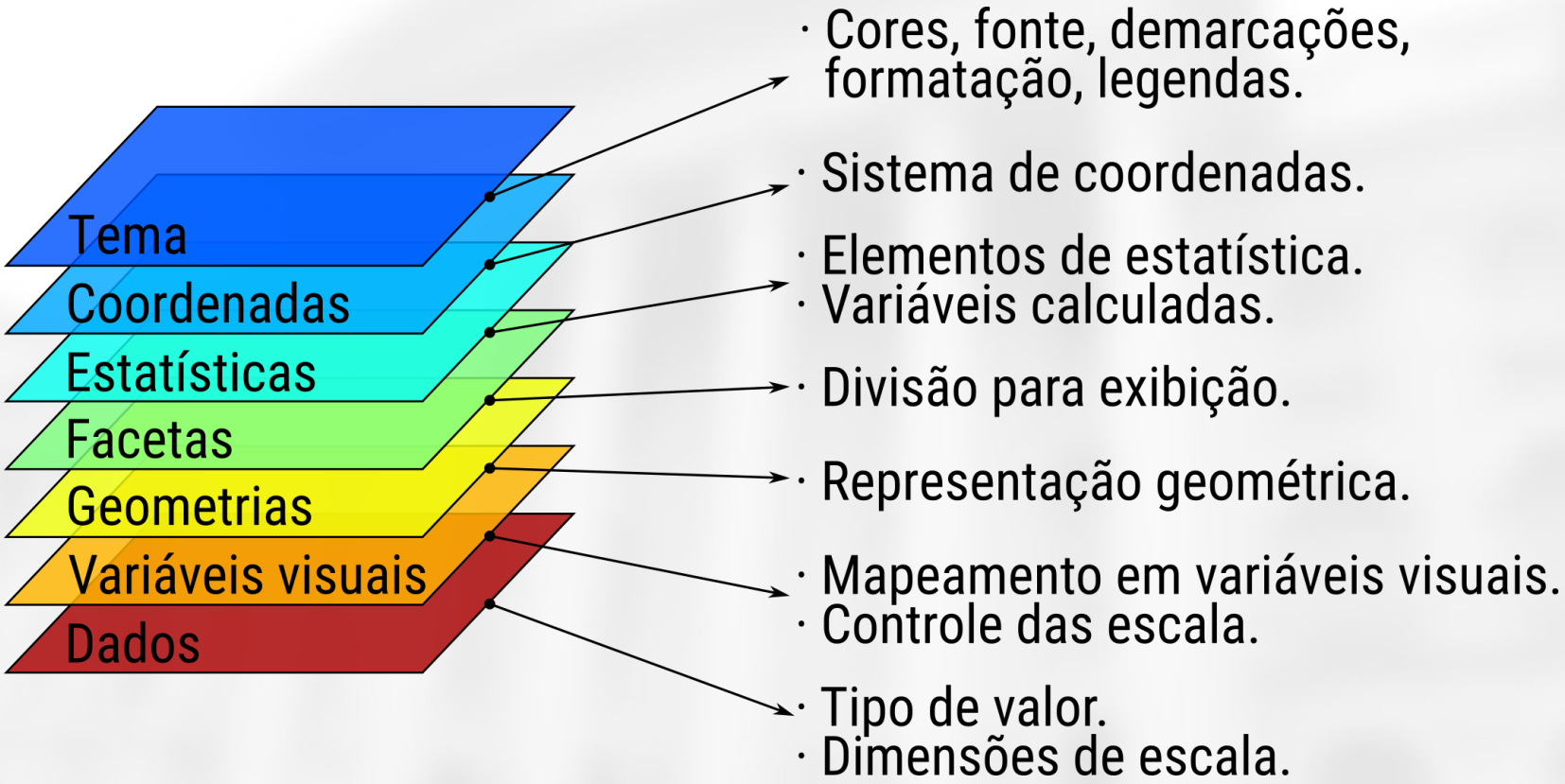

Figura 5. As camadas na gramática dos gráficos.

Camada 1: dados

Data: tb

	name	age	cm	kg	bmi	position	games	goal	ass	yel
1	Sadio Mané	21	175	69	22.53061	M	9	4	3	1
2	Kevin Kampl	23	180	63	19.44444	M	9	2	4	2
3	Alan	24	182	73	22.03840	FW	9	4	2	0
4	André Ramalho	22	182	77	23.24598	D	9	1	NA	4
5	Stefan Hierländer	23	180	71	21.91358	M	2	0	NA	0
6	Christoph Leitgeb	28	172	69	23.32342	M	8	0	1	3
7	Jonatan Soriano	28	180	71	21.91358	FW	7	8	1	1
8	Stefan Ilsanker	24	186	74	21.38976	M	9	0	NA	1
9	Martin Hinteregger	21	184	78	23.03875	D	9	0	NA	3
10	Andreas Ulmer	28	175	69	22.53061	D	8	0	NA	0
11	Christian Schwegler	29	177	73	23.30109	D	8	0	1	3

Figura 6. A camada dos dados.

Camada 1: dados

Deve se estar atento ao tipo de valor/objeto.

- Quantitativa: variável numérica discreta ou contínua.
- Qualitativa: variável nominal ou ordinal.
- Cronológica: variável de data ou data-tempo.
- Geográfica: objeto como polígonos, por exemplo.

Camada 2: mapeamento em variáveis visuais

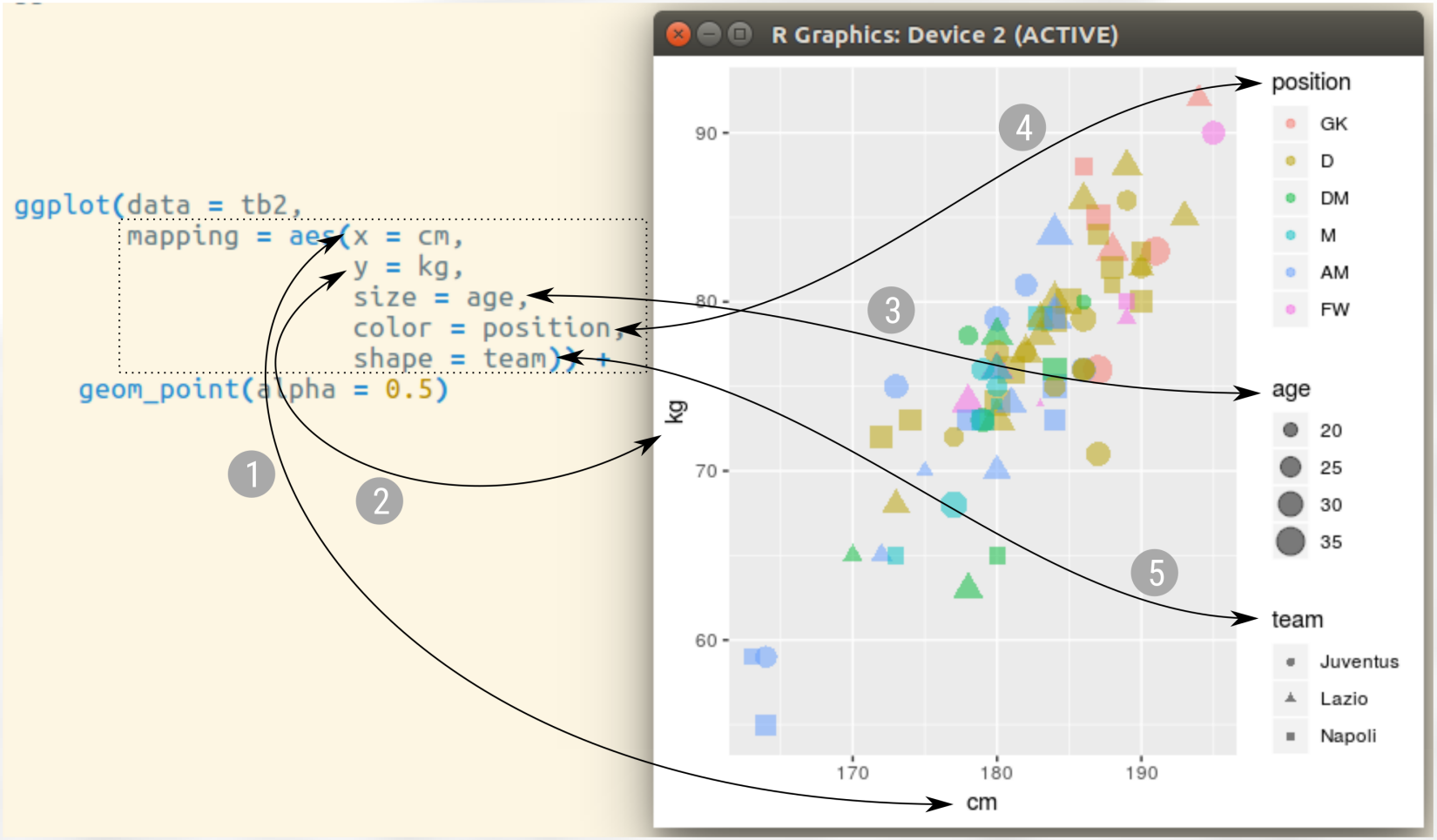

Figura 7. A camada de mapeamento dos valores em variáveis visuais.

Camada 2: mapeamento em variáveis visuais

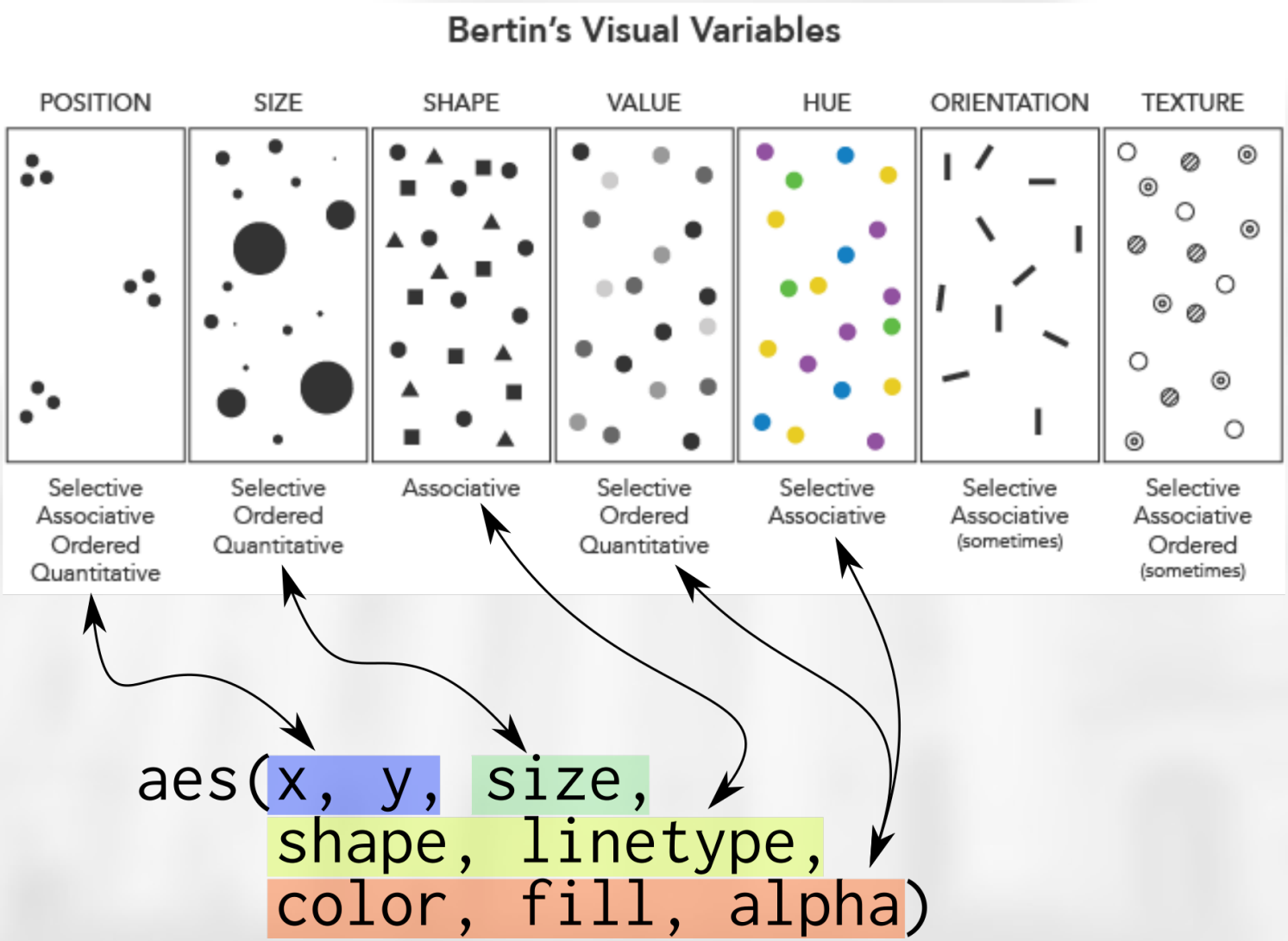

Figura 8. Variáveis visuais disponíveis no `ggplot2`.

Camada 2: mapeamento em variáveis visuais

Funções para controle de escala:

```
scale_alpha_:
  continuous date datetime discrete identity manual ordinal
scale_colour_:
  brewer continuous date datetime discrete distiller gradient
  gradient2 gradientn grey hue identity manual ordinal
scale_continuous_:
  identity
scale_discrete_:
  identity manual
scale_fill_:
  brewer continuous date datetime discrete distiller gradient
  gradient2 gradientn grey hue identity manual ordinal
scale_linetype_:
  continuous discrete identity manual
scale_shape_:
  continuous discrete identity manual ordinal
scale_size_:
  area continuous date datetime discrete identity manual ordinal
scale_x_:
  continuous date datetime discrete log10 reverse sqrt time
scale_y_:
  continuous date datetime discrete log10 reverse sqrt time
```

Camada 3: geometrias

```
ggplot(data = tb3,
       mapping = aes(x = position,
                     y = goal,
                     color = team)) +
    geom_point() +
    geom_line(mapping = aes(group = team))

ggplot(data = tb3,
       mapping = aes(x = position,
                     y = goal,
                     fill = team)) +
    geom_col(position = "dodge")
```

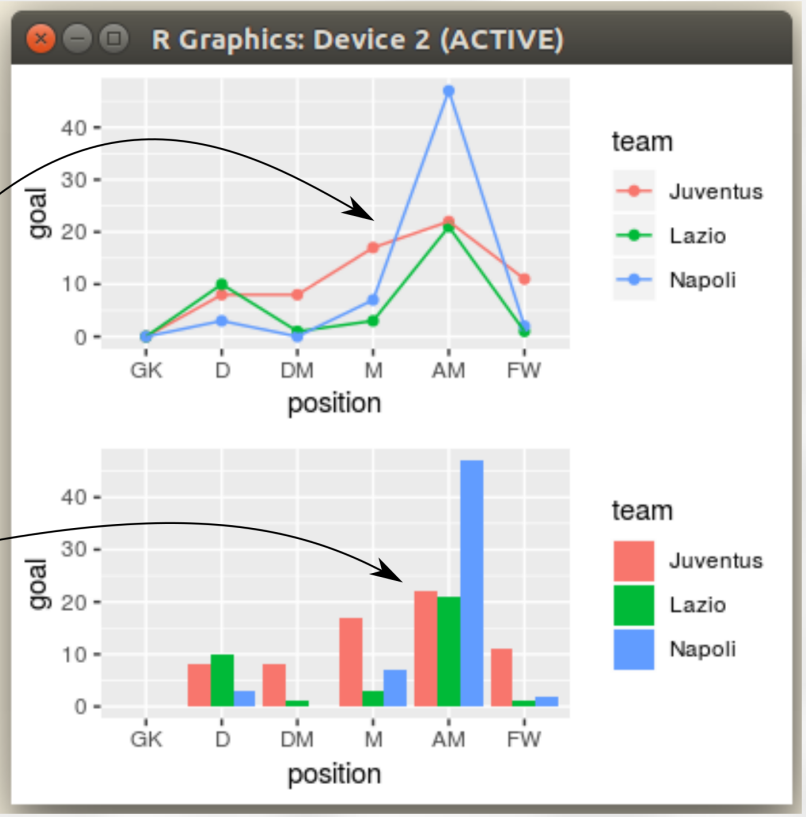

Figura 9. A camada de elementos geométricos.

Camada 3: geometrias

Funções disponíveis para a camada de geometria:

```
geom_abline          geom_freqpoly        geom_rect
geom_area            geom_hex             geom_ribbon
geom_bar             geom_histogram       geom_rug
geom_bin2d           geom_hline           geom_segment
geom_blank           geom_jitter          geom_sf
geom_boxplot         geom_label           geom_sf_label
geom_col             geom_line            geom_sf_text
geom_contour         geom_linerange       geom_smooth
geom_count           geom_map             geom_spoke
geom_crossbar        geom_path            geom_step
geom_curve           geom_point           geom_text
geom_density         geom_pointrange      geom_tile
geom_density2d       geom_polygon         geom_violin
geom_density_2d      geom_qq              geom_vline
geom_dotplot         geom_qq_line         update_geom_defaults
geom_errorbar        geom_quantile
geom_errorbarh       geom_raster
```

Camada 4: divisão em facetas

```
ggplot(data = tb2,
       mapping = aes(x = cm,
                     y = kg)) +
    geom_point(pch = 1) +
    facet_wrap(facets = ~position,
               nrow = 3)
```

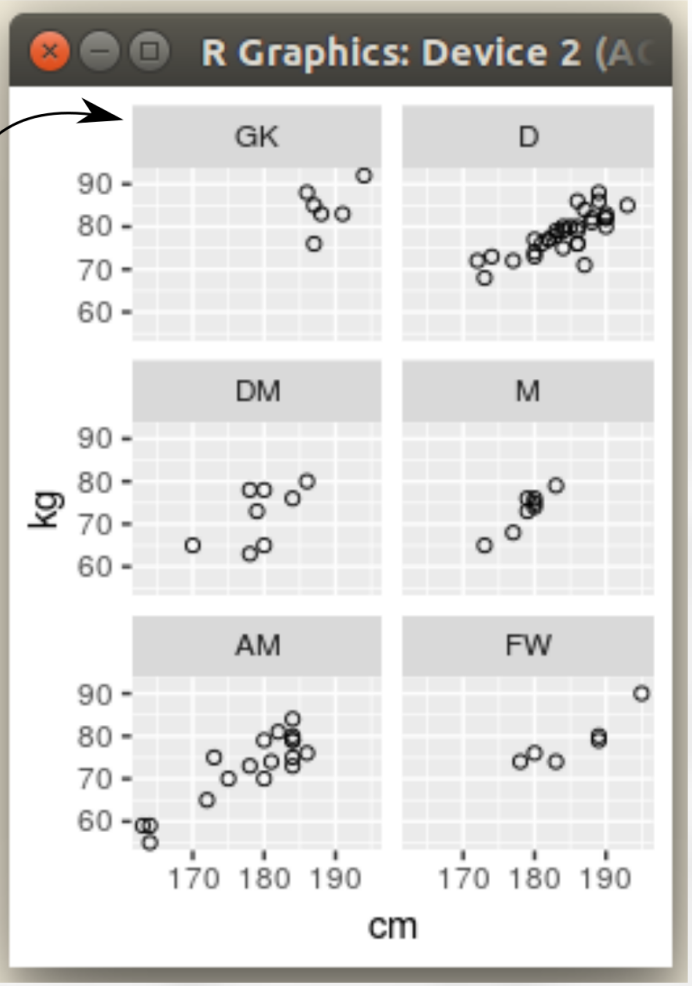

Figura 10. A camada da divisão em facetas.

Camada 4: divisão em facetas

Funções disponíveis para divisão em facetas:

```
facet_grid
facet_null
facet_wrap
```

Camada 5: estatística

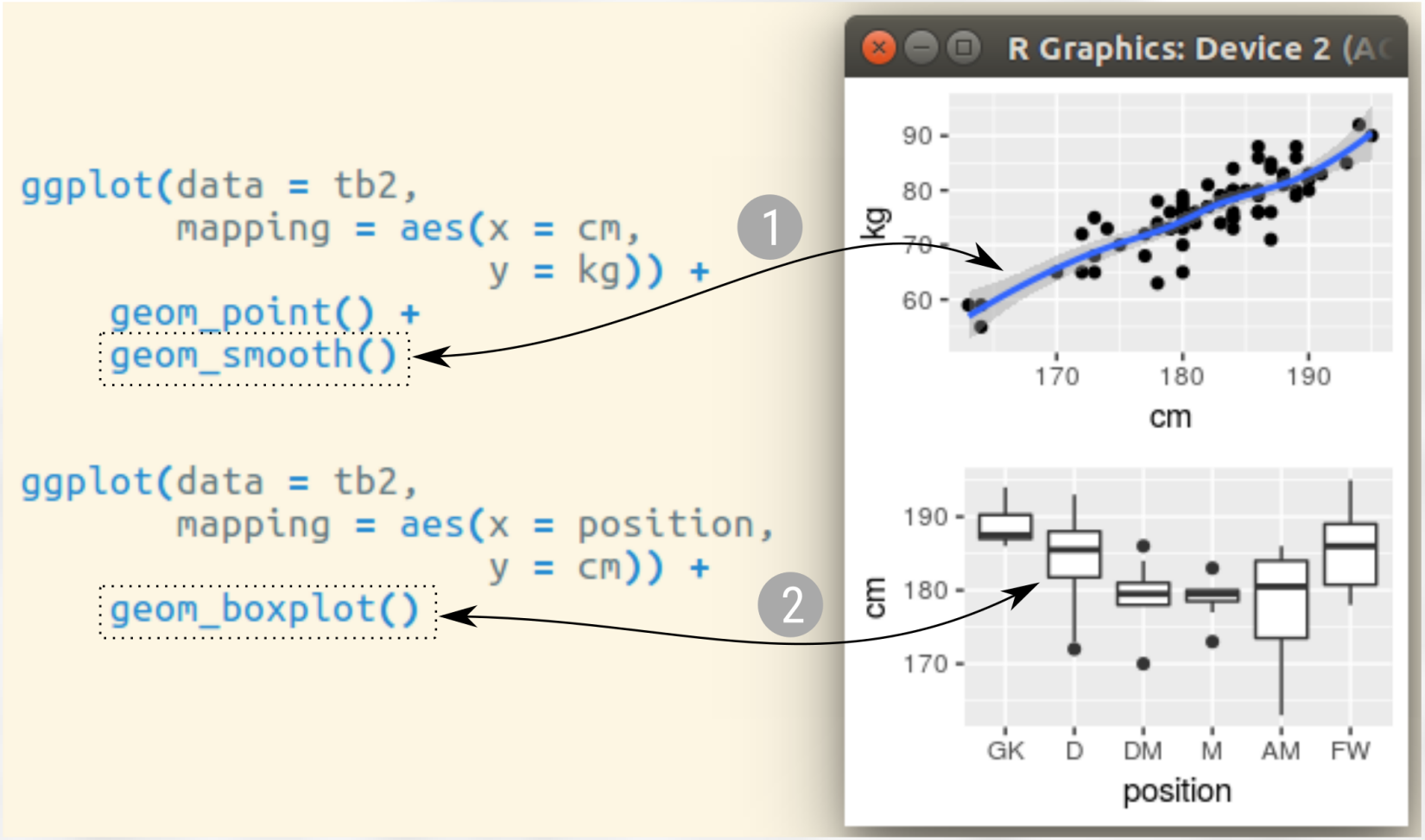

Figura 11. A camada de elementos de estatística

Camada 5: estatística

Funções disponíveis para a camada estatística:

```
stat_bin              stat_ecdf             stat_sum
stat_bin2d            stat_ellipse          stat_summary
stat_bin_2d           stat_function         stat_summary2d
stat_binhex           stat_identity         stat_summary_2d
stat_bin_hex          stat_qq               stat_summary_bin
stat_boxplot          stat_qq_line          stat_summary_hex
stat_contour          stat_quantile         stat_unique
stat_count            stat_sf               stat_ydensity
stat_density          stat_sf_coordinates   update_stat_defaults
stat_density2d        stat_smooth
stat_density_2d       stat_spoke
```

Camada 6: coordenadas

```
ggplot(data = tb4,
       mapping = aes(x = 0,
                     y = goal,
                     fill = position)) +
    geom_col(color = "black")


ggplot(data = tb4,
       mapping = aes(x = 0,
                     y = goal,
                     fill = position)) +
    geom_col(color = "black") +
    coord_polar(theta = "y")
```

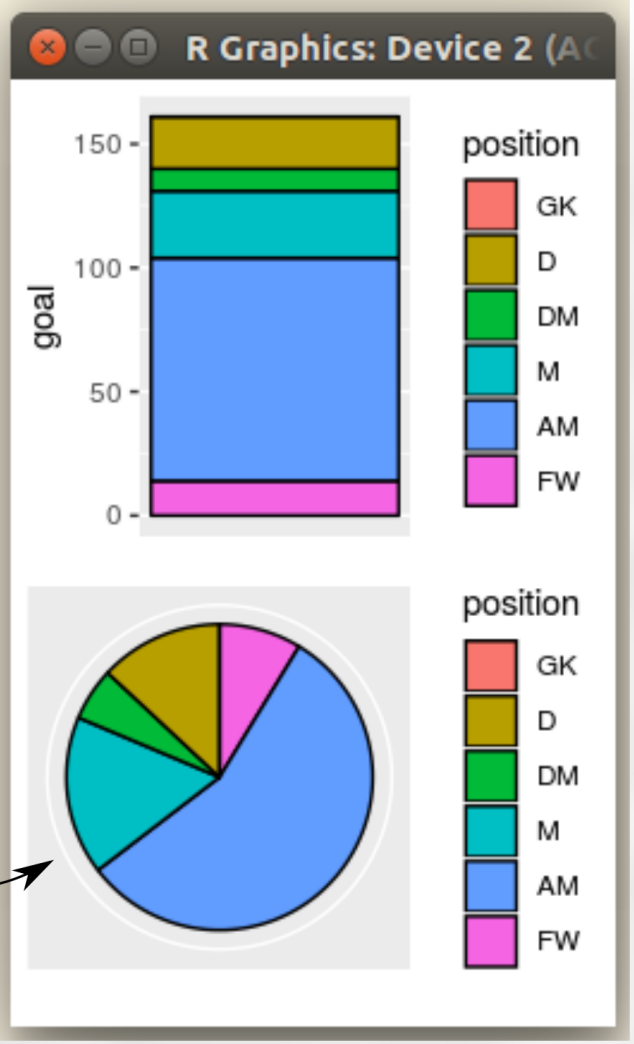

Figura 12. A camada do sistema de coordenadas.

Camada 6: coordenadas

Funções disponíveis para a camada de coordenadas:

```
coord_cartesian
coord_equal
coord_fixed
coord_flip
coord_map
coord_munch
coord_polar
coord_quickmap
coord_sf
coord_trans
```

Camada 7: tema

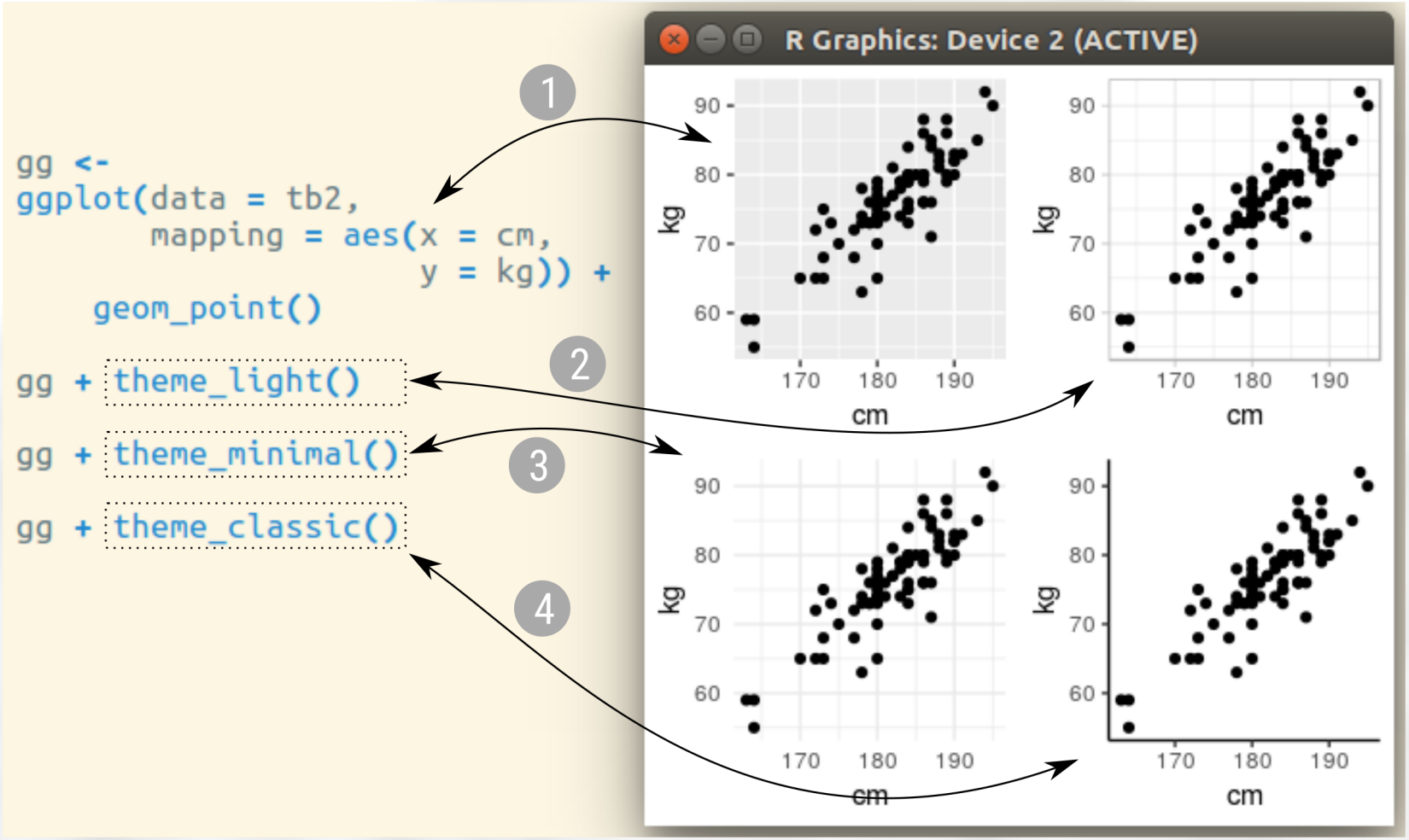

Figura 13. A camada de tema.

Camada 7: tema

Funções disponíveis para a camada de tema:

```
theme_bw
theme_classic
theme_dark
theme_get
theme_gray
theme_grey
theme_light
theme_linedraw
theme_minimal
theme_replace
theme_set
theme_test
theme_update
theme_void
```

```
TODO WALMES FIXME e quais as funções que sobram?
ls_ggplot %>%
    str_subset("[[:upper:]]", negate = TRUE)
```

Data Visualization with ggplot2 : : CHEAT SHEET

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and geoms—visual marks that represent data points.

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +                                   required
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT> , position = <POSITION> ) +                 Not required, sensible defaults supplied
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings | data | geom

qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank()
(Useful for expanding limits)

b + geom_curve(aes(yend = lat + 1, xend=long+1,curvature=z)**)** - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend="butt", linejoin="round", linemitre=1**)**
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(group = group)**)**
x, y, alpha, color, fill, group, linetype, size

b + geom_rect(aes(xmin = long, ymin=lat, xmax= long + 1, ymax = lat + 1)**)** - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin=unemploy - 900, ymax=unemploy + 900)**)** - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept=0, slope=1)**)**
b + geom_hline(aes(yintercept = lat)**)**
b + geom_vline(aes(xintercept = long)**)**

b + geom_segment(aes(yend=lat+1, xend=long+1)**)**
b + geom_spoke(aes(angle = 1:1155, radius = 1)**)**

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin"))
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly() x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5**)** x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)**)** x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fl))

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

continuous x , continuous y

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter(height = 2, width = 2**)**
x, y, alpha, color, fill, shape, size

e + geom_point(), x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile(), x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)**, x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)**, x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)**, x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

discrete x , continuous y

f <- ggplot(mpg, aes(class, hwy))

f + geom_col(), x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center"**)**, x, y, alpha, color, fill, group

f + geom_violin(scale = "area"**)**, x, y, alpha, color, fill, group, linetype, size, weight

discrete x , discrete y

g <- ggplot(diamonds, aes(cut, color))

g + geom_count(), x, y, alpha, color, fill, shape, size, stroke

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500)**)**
x, y, alpha, color, fill, linetype, size, weight

h + geom_density2d()
x, y, alpha, colour, group, linetype, size

h + geom_hex()
x, y, alpha, colour, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

j + geom_crossbar(fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar(), x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map**)**
+ expand_limits(x = map$long, y = map$lat**)**,
map_id, alpha, color, fill, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, colour, group, linetype, size, weight

l + geom_raster(aes(fill = z)**)**, hjust=0.5, vjust=0.5, interpolate=FALSE)
x, y, alpha, fill

l + geom_tile(aes(fill = z)**)**, x, y, alpha, color, fill, linetype, size, width

Figura 14. Cartão de referência dos recursos do `ggplot2`.

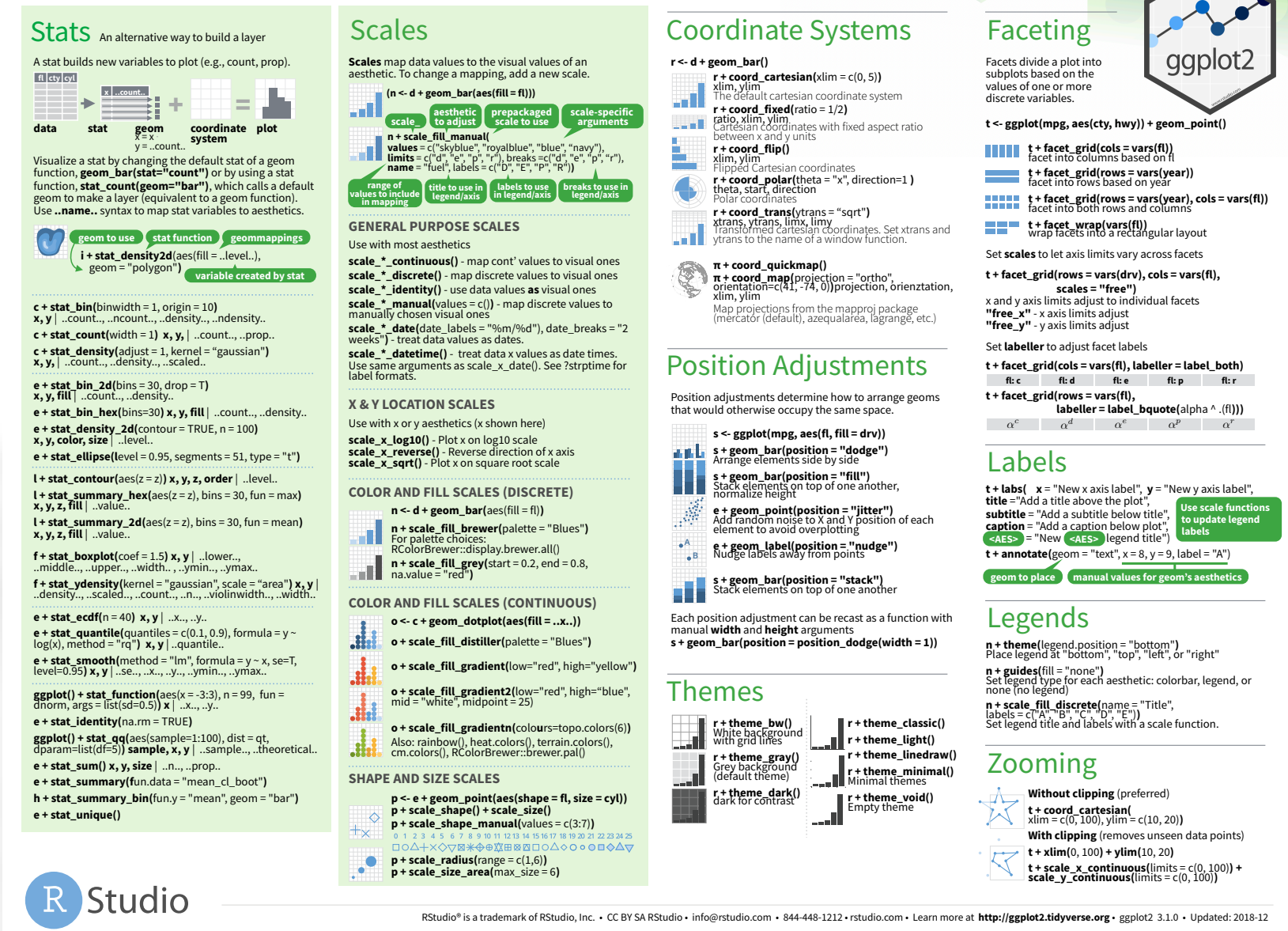

Figura 15. Cartão de referência dos recursos do `ggplot2`.